**MINI PROJECT- 6**

**CS 6313.001 - Statistical Methods for Data Science**

**FALAK SINGHAL  
(fxs161530@utdallas.edu)**

**Section-1**

1. Bias and standard error of the dataset as found by the bootstrap implementation is found to be –

Bias Value: -0.005501147

Standard Error: 0.0962783

1. 2.5th and 97.5th Percentiles of the Distribution are –

[3.679671, 4.063312]

1. 2.5th and 97.5th Percentiles ofare –

[-0.1963798, 0.1872621]

1. Confidence Intervals –
2. Normal Approximation

[3.692849, 4.070253]

1. Basic Bootstrap

[3.688788, 4.072430]

1. Percentile Bootstrap

[3.679671 4.063312]

Following are the results obtained from boot package automated analysis –

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = cpu, statistic = log.mean, R = 1000, sim = "ordinary",

stype = "i")

Bootstrap Statistics :

original bias std. error

t1\* 3.87605 -0.007426565 0.09801347

> boot.ci(npar.boot)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

boot.ci(boot.out = npar.boot)

Intervals :

Level Normal Basic

95% ( 3.691, 4.076 ) ( 3.704, 4.089 )

Level Percentile BCa

95% ( 3.663, 4.048 ) ( 3.691, 4.073 )

Based on the automated analysis, it can be inferred that the results obtained by the custom procedures are accurate.

**Section-2**

require(boot)

#function to get N bootstrap samples

getNBootStrapSamples<-function(data, no.Of.Resamples, sample.size){

samples<-replicate(n = no.Of.Resamples,sample(x=data,size = sample.size,replace = TRUE));

return(samples);# samples will be in column binded form

}

getBootStrapDistribution<-function(data,no.Of.Resamples, sample.size){

#get Samples

samples<-getNBootStrapSamples(data,no.Of.Resamples,sample.size);

#apply log(mean(x)) column wise, essentially getting bootstrap distribution of theta.hat

bootstrapDistribution<-apply(X = samples,MARGIN = 2,FUN = function(x) log(mean(x)))

return(bootstrapDistribution)

}

calculateBiasAndSE<-function(bootStrapDistribution, original.data){

#bias\* = E(theta\*.hat) - theta.hat

#where theta\*.hat is bootstrap Distribution of theta.hat

bias<-mean(bootStrapDistribution)-log(mean(original.data))

se <- sd(bootStrapDistribution)

return(list(bias=bias, standardError=se))

}

solveProject6<-function(){

require(boot)

cpu<-scan(file = "cpu.txt");

set.seed(2008);

cpu.size<-length(cpu)

#boot strap distribution of Theta Hat

bsd<-getBootStrapDistribution(data = cpu,no.Of.Resamples = 1000,sample.size = cpu.size)

#part(a)

biasAndSE<-calculateBiasAndSE(bootStrapDistribution = bsd,original.data = cpu)

#part(b)

#alpha = 0.05

#calculate lower and upper percentiles from bootstrap distribution

lower<-0.025\*(length(bsd)+1)

upper<-0.975\*(length(bsd)+1)

percentiles1<-sort(bsd)[c(lower,upper)]

#part(c)

#2.5th and 97.5th percentiles of theta.hat - theta = percentile of theta\*.hat-theta.hat

logmean<-log(mean(cpu))

percentiles2<-percentiles1-logmean

#CI1 = Normal BS CI

CI.normal<-c(logmean-biasAndSE$bias-qnorm(0.975)\*biasAndSE$standardError,logmean-biasAndSE$bias-qnorm(0.025)\*biasAndSE$standardError)

#CI2 : percentiles method

CI.percent<-percentiles1

#CI3: basic

CI.basic<-c(2\*logmean-percentiles1[2], 2\*logmean-percentiles1[1])

return(list(bias=biasAndSE$bias, StandardError=biasAndSE$standardError,Percentiles.Of.Theta.hat=percentiles1, Percentiles.Of.Theta.hat.minus.Theta=percentiles2,Basic.CI=CI.basic,NormalApproximation.CI=CI.normal, Percentile.CI=CI.percent))

}

#test

log.mean <- function(x,indices) {

result <- log(mean(x[indices]))

return(result)

}

npar.boot<-boot(data = cpu,statistic = log.mean,R = 1000,sim = "ordinary", stype = "i")